```python


from a0_items import *
from plg_User_agent import *

from Proxy_List_Scrapper import Scrapper, Proxy, ScrapperException

from fp.fp import FreeProxy

## importing socket module
import socket


################################################################################################################################

# https://www.tutorialspoint.com/python-program-to-find-the-ip-address-of-the-client
def get_my_IP_and_HOST ():

    ## getting the hostname by socket.gethostname() method
    hostname = socket.gethostname()

    ## getting the IP address using socket.gethostbyname() method
    ip_address = socket.gethostbyname(hostname)

    ## printing the hostname and ip_address
    print(f"Hostname: {hostname}")
    print(f"IP Address: {ip_address}")

    return ip_address, hostname


################################################################################################################################

################################################################################################################################

# https://pypi.org/project/Proxy-List-Scrapper/
def ALL_proxies ():

    SSL = 'https://www.sslproxies.org/',
    GOOGLE = 'https://www.google-proxy.net/',
    ANANY = 'https://free-proxy-list.net/anonymous-proxy.html',
    UK = 'https://free-proxy-list.net/uk-proxy.html',
    US = 'https://www.us-proxy.org/',
    NEW = 'https://free-proxy-list.net/',
    SPYS_ME = 'http://spys.me/proxy.txt',
    PROXYSCRAPE =
    'https://api.proxyscrape.com/?request=getproxies&proxytype=all&country=all&ssl=all&ano
    nymity=all',
    PROXYNOVA = 'https://www.proxynova.com/proxy-server-list/'
    PROXYLIST_DOWNLOAD_HTTP = 'https://www.proxy-list.download/HTTP'
    PROXYLIST_DOWNLOAD_HTTPS = 'https://www.proxy-list.download/HTTPS'
    PROXYLIST_DOWNLOAD_SOCKS4 = 'https://www.proxy-list.download/SOCKS4'
    PROXYLIST_DOWNLOAD_SOCKS5 = 'https://www.proxy-list.download/SOCKS5'
    ALL = 'ALL'

    return ALL


################################################################################################################################

def get_ALL_proxies ():

```

```
    LIST_proxy = []

    scrapper = Scrapper(category = ALL_proxies(), print_err_trace=False)

    # Get ALL Proxies According to your Choice
    data = scrapper.getProxies()

    # Print These Scrapped Proxies
    #print("Scrapped Proxies:")

    for item in data.proxies:
       #print('{}:{}'.format(item.ip, item.port))

       LIST_proxy.append ('{}:{}'.format(item.ip, item.port))

    # Print the size of proxies scrapped
    print("Total Proxies --> " + str(data.len))

    # Print the Category of proxy from which you scrapped
    print("Category of the Proxy -- > " + str(data.category))

    return LIST_proxy


#print (get_ALL_proxies ())



############################################################################################################################################

# https://proxyscrape.com/blog/how-to-make-a-proxy-checker-in-python
import urllib.request , socket


socket.setdefaulttimeout(180)

def is_bad_proxy(pip):

    try:

       proxy_handler = urllib.request.ProxyHandler({'http': pip})

       opener = urllib.request.build_opener(proxy_handler)
       #opener.addheaders = [('User-agent', 'Mozilla/5.0')]
       opener.addheaders = [('User-agent', get_random_USER_AGENT ()[1])]

       urllib.request.install_opener(opener)

       sock=urllib.request.urlopen(url = 'http://www.google.com')

    except urllib.error.HTTPError as e:

       print('Error code: ', e.code)
       return e.code

    except Exception as detail:

       print( "ERROR:", detail)
       return 1

    return 0



def get_GOOD_proxies ():

```

```
    LIST_good_proxy = []
    LIST_bad_proxy = []

    LIST_proxy = get_ALL_proxies ()

    for item in LIST_proxy:

       if is_bad_proxy(item): #using google.com to check

          print ("NOT working --> " + str(item), flush=True)
          LIST_bad_proxy.append (item)

       else:

         print ("Working --> " + str(item), flush=True)
         LIST_good_proxy.append (item)

    return LIST_good_proxy, LIST_bad_proxy, len(LIST_good_proxy), len(LIST_bad_proxy)


#print ("ALL GOOD PROXIES -->" + str(get_GOOD_proxies()[2]), flush=True)



################################################################################################################################################


def check_proxy_request (url, LIST_proxy, LIST_index):

    try:

       proxy = {"http": LIST_proxy[LIST_index], "https": LIST_proxy[LIST_index]}
       response = requests.get(url, proxies=proxy, headers= get_random_USER_AGENT ()[0])
       # proxies is a mandatory parameter name.

       #print("Proxy used --> " + str(LIST_proxy[LIST_index]), flush=True)
       #print("Response Status Code --> " + str (response.status_code), flush=True)
       #print("Response data in Text format --> " + str( response.text), flush=True)
       #print("Response data in JSON format --> " + str( response.json()), flush=True)

       proxy_response = str(response.text) + " --> " + str(response.status_code)

       return proxy_response #mulitple return values will output 'None type error' in try...except statement

    except:

       pass



################################################################################################################################################


# THIS COULD BE THE FUNCTION WE NEED !!!

# https://willdrevo.com/using-a-proxy-with-a-randomized-user-agent-in-python-requests

# ADD USER AGENT

def test_proxies_in_loop ():

    LIST_P = get_ALL_proxies ()
    #print (len(LIST_P))

    LIST_proxy_GOOD = []
    LIST_proxy_BAD = []
```

```
191
192      my_ip = '122.151.43.108'
193
194      for counter, LIST_I in enumerate(LIST_P):
195
196         proxy_response = check_proxy_request (url ="http://httpbin.org/ip", LIST_proxy=
         LIST_P, LIST_index=counter)
197         #print (proxy_response, flush=True)
198
199         # if proxy response is None, or my ip appears in the response ip list, and
         response error code is NOT 200
200         if proxy_response is None or '200' not in str(proxy_response) or my_ip in
         proxy_response:
201
202            print (my_ip + " in --> " + str(proxy_response), flush=True)
203            LIST_proxy_BAD.append(proxy_response)
204
205         else:
206
207            print (my_ip + " NOT in --> " + str(proxy_response), flush=True)
208            LIST_proxy_GOOD.append(proxy_response)
209            #pass
210
211      return LIST_proxy_GOOD, LIST_proxy_BAD
212
213   #test_proxies_in_loop ()
214
215   #print ("GOOD proxies --> " + test_proxies_in_loop ()[0])
216   #print ("BAD proxies --> " + test_proxies_in_loop ()[1])
217
218
219   ##########################################################################################
   ##############3
220
221   # https://oxylabs.io/resources/integrations/python-requests
222   def check_proxies_request ():
223
224      proxies = get_ALL_proxies ()
225
226      for index in range(len(proxies)):
227
228         try:
229            proxy = {"http": proxies[index], "https": proxies[index]}
230            response = requests.get(url="http://httpbin.org/ip", proxies=proxy, headers=
            get_random_USER_AGENT ()) # proxies is a mandatory parameter name.
231
232            print("Proxy used --> " + str(proxies[index]), flush=True)
233            print("Response Status Code --> " + str (response.status_code), flush=True)
234            print("Response data in Text format --> " + str( response.text), flush=True)
235
236         except:
237
238            pass
239
240
241   #check_proxies_request ()
242
243
244   ##########################################################################################
   ########
245
246
247   def check_proxies_request_1 ():
248
249      LIST_proxy_GOOD = []
250      LIST_proxy_BAD = []
251
```

```
252      my_ip = '122.151.43.108'
253
254      LIST_proxy = get_ALL_proxies ()
255
256      for index in range(len(LIST_proxy)):
257
258         try:
259            proxy = {"http": LIST_proxy[index], "https": LIST_proxy[index]}
260            response = requests.get(url ="http://httpbin.org/ip", proxies=proxy, headers=
               get_random_USER_AGENT ()[0]) # proxies is a mandatory parameter name.
261
262            #print("Proxy used --> " + str(LIST_proxy[index]), flush=True)
263            #print("Response Status Code --> " + str (response.status_code), flush=True)
264            #print("Response data in Text format --> " + str( response.text), flush=True)
265
266            proxy_response_TEXT =  response.text
267            proxy_response_CODE = response.status_code
268
269            # if my ip appears in the response ip list, or response error code is NOT
               200
270            if my_ip in proxy_response_TEXT or '200' not in str(proxy_response_CODE):
271
272               print (my_ip + " in --> " + str(proxy_response_TEXT), flush=True)
273               print (str(proxy_response_CODE), flush=True)
274               LIST_proxy_BAD.append(proxy_response_TEXT)
275
276            else:
277
278               print (my_ip + " NOT in --> " + str(proxy_response_TEXT), flush=True)
279               print (str(proxy_response_CODE), flush=True)
280               LIST_proxy_GOOD.append(proxy_response_TEXT)
281
282         except:
283
284            pass
285
286      return LIST_proxy_BAD, LIST_proxy_GOOD
287
288
289   #check_proxies_request_1 ()
290
291   #print ("BAD IP list --> " + check_proxies_request_1()[0], flush=True)
292
293   #print ("GOOD IP list --> " + check_proxies_request_1()[1], flush=True)
294
295
296
297   ##########################################################################
298   ###### Use the free proxy library --> https://pypi.org/project/free-proxy/
299   ##########################################################################
300
301   def get_ALL_FREE_proxy_url ( COUNT_proxy):
302
303      LIST_proxy_url = []
304
305      for x in range (COUNT_proxy):
306
307         proxy = FreeProxy( timeout=1).get()
308
309         LIST_proxy_url.append(proxy)
310         #print (proxy, flush=True)
311
312
313      return LIST_proxy_url
314
315
316   #LIST_country = ['US', 'BR']
```

```python
#get_ALL_FREE_proxy_url ( COUNT_proxy = 50)


#https://codepal.ai/code-generator/query/FWUNszND/python-function-access-unblocked-google
# Use Google to check if the proxy url works
def access_unblocked_google(proxy_url):
    """
    Function to access unblocked Google using a proxy server.

    This function sends a GET request to a proxy server, which then forwards the request to Google.
    By using a proxy server, we can bypass any restrictions or blocks that may be in place.

    Returns:
    - str:
        The HTML content of the Google homepage.

    Raises:
    - requests.exceptions.RequestException:
        If there is an error while making the request to the proxy server or Google.
    """


    # Define the proxy server URL
    #proxy_url = "http://proxy.example.com"

    try:
        # Send a GET request to the proxy server with the Google URL
        response = requests.get(proxy_url, params={"url": "https://www.google.com"})

        # Check if the request was successful (status code 200)
        if response.status_code == 200:

            # Return the HTML content of the Google homepage
            return response.text

        else:

            # Raise an exception if the request was not successful
            response.raise_for_status()

    except requests.exceptions.RequestException as e:

        # Raise an exception if there is an error while making the request
        raise e



LIST_proxy_url =  get_ALL_FREE_proxy_url ( COUNT_proxy=50)
# Example usage of the access_unblocked_google function

for proxy_url in LIST_proxy_url:

    try:
        google_html = access_unblocked_google(proxy_url=proxy_url)
        print(google_html, flush=True)

    except requests.exceptions.RequestException as e:

         print(f"Error accessing unblocked Google: {e}", flush=True)

```